**Discussion and Solution of the Problem**

**Ujjwal Pandya**

- Arrays and Linked Lists both are linear data structures, but they both have some advantages and disadvantages over each other. One advantage of the linked list is that elements can be added to it indefinitely, where an array will eventually get filled or have to be resized. Elements are also easily removed from a linked list, whereas removing elements from an array leaves empty spaces that are a waste of computer memory. However, unlike arrays that allow random access to the elements contained within them, a link list only allows sequential access to its elements. Linked lists also use more storage space in a computer's memory as each node in the list contains both a data item and a reference to the next node. It follows that linked lists should be used for large lists of data where the total number of items in the list is changing. Arrays, on the other hand, are better suited to small lists, where the maximum number of items that could be on the list is known.

**Annanya Mittal**

- An array is a data structure that contains a collection of similar data elements where as the linked list is considered as non primitive data structure which contains a collection of unordered linked elements known as nodes

**Akshat:**

A linked list is a data structure that can hold an arbitrary number of data items and can easily change the size to add or remove items. A linked list, at its simplest, is a pointer to a data node. Each data node is then composed of data (possibly a record with several data values), and a pointer to the next node. At the end of the list, the pointer is set to null.

By nature of its design, a linked list is great for storing data when the number of items is either unknown or subject to change. However, it provides no way to access an arbitrary item from the list, short of starting at the beginning and traversing through every node until you reach the one you want. The same is true if you want to insert a new node at a specific location. It is not difficult to see the problem of inefficiency.

**Somesh Anjana -**

In computer science, a linked list is a linear collection of data elements whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence.

Singly linked lists contain nodes which have a data field as well as 'next' field, which points to the next node in line of nodes. in the below list first value is points toward the next value.

Singly-linked-list.svg

**Jeel:-**

The linked list is a sequence of data elements, which are connected together via links. Each data element contains a connection to another data element in the form of a pointer. Python does not have linked lists in its standard library. ... In this type of data structure, there is only one link between any two data elements.

However, unlike arrays which allow random access to the elements contained within them, a link list only allows sequential access to its elements. Linked lists also use more storage space in a computer's memory as each node in the list contains both a data item and a reference to the next node.

**Muskan:**

Linked lists are linear data structures consisting of nodes. Nodes are the individual objects which holds the data and reference to the next node in the list.

2 --> 4 --> 3 --> null

This is a linked list, in this case 2 the first element points towards the next node that contains 4 (integer) and the reference to the next node and so on. And 3 (integer) points towards the null node i.e. the list ends there. One can efficiently delete and insert in the linked list.

The problem discussed in the group is about checking the cycle of the linked list.

We need to check the index positions of any of the element w.r.t. to another index position, to check if there is a cycle going on or the end of the list.

**Shambhavi Goswami**

A linked list is a linear data structure where each element is a separate object.

Each element is referred as a node of a list is comprising of two items - the data and a reference to the next node. Also, it is a dynamic data structure. The number of nodes in a list is not fixed and it can grow and shrink as per need.

Workflow for the problem,

Our approach will be going through each node one by one to check if the node’s reference exists, the list will be a cyclic. Then, we need to check for the starting node of the cycle.

* For detecting the cycle, we will use pointers and let them initially point to first node of linked list.
* Move the first pointer one node at a time and move the second pointer two nodes at a time.
* If it contains a cycle then after some time both pointers will enter in the cycle and will definitely meet at some node.
* The two pointers will meet only if there is cycle in the linked list

If the cycle exists, we are supposed to find the starting node of cycle .Let both the pointers move one node at a time. When they meet that will be starting node of cycle.

If arrays are used for the solving the above, it would posses a con, to store data that is ,arrays are static structures and hence cannot be easily extended or reduced to fit the data set.

**Abhijit Panchal**

A linked list is a data structure made of a chain of *node* objects. Each node contains a value and a pointer to the next node in the chain.

Linked lists are preferred over arrays due to their dynamic size and ease of insertion and deletion properties.

The head pointer points to the first node, and the last element of the list points to null. When the list is empty, the head pointer points to null.

**Ujjwal Pandya**

- Arrays and Linked Lists both are linear data structures, but they both have some advantages and disadvantages over each other. One advantage of the linked list is that elements can be added to it indefinitely, where an array will eventually get filled or have to be resized. Elements are also easily removed from a linked list, whereas removing elements from an array leaves empty spaces that are a waste of computer memory. However, unlike arrays that allow random access to the elements contained within them, a link list only allows sequential access to its elements. Linked lists also use more storage space in a computer's memory as each node in the list contains both a data item and a reference to the next node. It follows that linked lists should be used for large lists of data where the total number of items in the list is changing. Arrays, on the other hand, are better suited to small lists, where the maximum number of items that could be on the list is known.

**Annanya Mittal**

- An array is a data structure that contains a collection of similar data elements where as the linked list is considered as non primitive data structure which contains a collection of unordered linked elements known as nodes

**Akshat:**

A linked list is a data structure that can hold an arbitrary number of data items and can easily change the size to add or remove items. A linked list, at its simplest, is a pointer to a data node. Each data node is then composed of data (possibly a record with several data values), and a pointer to the next node. At the end of the list, the pointer is set to null.

By nature of its design, a linked list is great for storing data when the number of items is either unknown or subject to change. However, it provides no way to access an arbitrary item from the list, short of starting at the beginning and traversing through every node until you reach the one you want. The same is true if you want to insert a new node at a specific location. It is not difficult to see the problem of inefficiency.

**Somesh Anjana -**

In computer science, a linked list is a linear collection of data elements whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence.

Singly linked lists contain nodes which have a data field as well as 'next' field, which points to the next node in line of nodes. in the below list first value is points toward the next value.

Singly-linked-list.svg

**Jeel:-**

The linked list is a sequence of data elements, which are connected together via links. Each data element contains a connection to another data element in the form of a pointer. Python does not have linked lists in its standard library. ... In this type of data structure, there is only one link between any two data elements.

However, unlike arrays which allow random access to the elements contained within them, a link list only allows sequential access to its elements. Linked lists also use more storage space in a computer's memory as each node in the list contains both a data item and a reference to the next node.

**Muskan:**

Linked lists are linear data structures consisting of nodes. Nodes are the individual objects which holds the data and reference to the next node in the list.

2 --> 4 --> 3 --> null

This is a linked list, in this case 2 the first element points towards the next node that contains 4 (integer) and the reference to the next node and so on. And 3 (integer) points towards the null node i.e. the list ends there. One can efficiently delete and insert in the linked list.

The problem discussed in the group is about checking the cycle of the linked list.

We need to check the index positions of any of the element w.r.t. to another index position, to check if there is a cycle going on or the end of the list.

**Shambhavi Goswami**

A linked list is a linear data structure where each element is a separate object.

Each element is referred as a node of a list is comprising of two items - the data and a reference to the next node. Also, it is a dynamic data structure. The number of nodes in a list is not fixed and it can grow and shrink as per need.

Workflow for the problem,

Our approach will be going through each node one by one to check if the node’s reference exists, the list will be a cyclic. Then, we need to check for the starting node of the cycle.

* For detecting the cycle, we will use pointers and let them initially point to first node of linked list.
* Move the first pointer one node at a time and move the second pointer two nodes at a time.
* If it contains a cycle then after some time both pointers will enter in the cycle and will definitely meet at some node.
* The two pointers will meet only if there is cycle in the linked list

If the cycle exists, we are supposed to find the starting node of cycle .Let both the pointers move one node at a time. When they meet that will be starting node of cycle.

If arrays are used for the solving the above, it would posses a con, to store data that is ,arrays are static structures and hence cannot be easily extended or reduced to fit the data set.

**Abhijit Panchal**

A linked list is a data structure made of a chain of *node* objects. Each node contains a value and a pointer to the next node in the chain.

Linked lists are preferred over arrays due to their dynamic size and ease of insertion and deletion properties.

The head pointer points to the first node, and the last element of the list points to null. When the list is empty, the head pointer points to null.